Many Looks Before a Leap

Xiaotie Deng [1], **Evangelos Milios** [2], **Andy Mirzaian** [3]
[1] Department of Computer Science, City University of Hong Kong
Kowloon, Hong Kong
[2] Department of Computer Science, York University
North York, Ontario, Canada, M3J 1P3
deng@cs.cityu.edu.hk, {eem, andy}@cs.yorku.ca.

Abstract

We are interested in the problem of map construction of an unknown environment. Even with a complete map, navigation of real autonomous mobile robots is still conducted with imperfect information. This can be caused by various factors, such as imperfect sensors, imperfect mechanical control capabilities, and imperfect road conditions. Computational resource constraints for real time performance may also prevent us from collecting and processing complete information about the environment. A major problem caused by these issues is that accummulated errors in the robot's self-localization may become arbitrarily large as the robot moves on. Therefore, in addition to the map, we would still need a strategy to help robots recognize the environment along the way and to guide movements toward the destination. This is called the guidance problem by Levitt and Lawton (Levitt & Lawton 1990).

We discuss an approach for the guidance problem using imprecise data, in the context of world models that are represented by maps with qualitative/topological information for the global environment, and quantitative/geometric information for the local environment, as suggested by Levitt and Lawton (Levitt & Lawton 1990), Kuipers and Levitt (Kuipers & Levitt 1988).

We allow the robot to be equipped with a constant number of sensors. We consider three cost measures: mechanical, sensing, and computation in decreasing order of importance. We propose a number of algorithms with their associated cost measures using a veriety of sensor/measurement devices.

Introduction

Navigation and mapping of autonomous mobile robots involve a modelling problem and a strategy design problem. First of all, we would need a model to describe the environment where mobile robots operate. This model allows us to build a map which takes into consideration the necessary information for a robot to recognize its location through its sensors, to match its observations of the environment with its map, and to keep track of its movement toward the goal. To efficiently build this map, we need to design a strategy for the robot to make decisions on what kind of observations to make and where to move next for more information. Even to navigate from one location to another with the help of a map, the robot would have to rely on its observations on the environment to maneuver between obstacles and to locate itself on the map.

With many possible models for an environment, two extreme cases (both with some drawbacks) are widely accepted: the graph model and the geometric model. The graph model abstracts away metric information of the environment and retains only topological information. It represents important locations as nodes and routes between them as edges. The robot can move from one node to another through an edge between them (Deng & Papadimitriou 1990). The geometric model records all the necessary geometric details of the environment where the robot operates (Canny & Reif 1987; Deng, Kameda, & Papadimitriou 1991; Guibas, Motwani, & Raghavan 1992).

Even with the world model fixed, different robot perceptual abilities can still make a difference. A robot with the power of unambiguously recognizing each individual node will see the world differently from one with more limited recognition power, for example ability to only recognize some property of the nodes. For example, Rivest and Schapire consider a graph world where nodes are divided into a small number of classes, for example, white and black colours, and can only be recognized as such (Rivest & Schapire 1989).

The computational power of the robot is also important. Blum and Hewitt consider a robot with the power of an automaton (Blum & Hewitt 1967). While a single automaton cannot search all mazes (Budach 1978), this is possible with two automata (Blum & Kozen 1978). In experimental mobile robotics (Cox &

Wilfong 1990), robots typically are Turing-equivalent (carrying general purpose computers on board).

Some models allow the robot to carry pebbles to distinguish a location from another. Blum and Kozen show that two pebbles would be enough to aid one automaton to search a maze (Blum & Kozen 1978). Deng and Mirzaian consider a robot which recognizes its own footprints but cannot distinguish/erase footprints (Deng & Mirzaian 1996).

More comprehensive world models, suggested by Levitt and Lawton (Levitt & Lawton 1990), Kuipers and Levitt (Kuipers & Levitt 1988), consider a cognitive map which provides qualitative/topological information for the global environment, and quantitative/geometric information for the local environment. Dudek et al. discuss a topological graph world model for which the incident edges at each node are circularly ordered (Dudek *et al.* 1991).

In line with the approach suggested by the world models of Levitt and Lawton (Levitt & Lawton 1990), Kuipers and Levitt (Kuipers & Levitt 1988), we may assume two distinguished modes for robots: stationary and moving. In the *stationary* mode, the robot can make precise measurement of the local environment using its sensors. In the *moving* mode, only qualitative changes in the environment can be observed. To avoid cumulative error in its self-localization, we require that the robot be able to infer precise geometric information about the environment only from a constant number (independent of the input size or complexity of the environment) of data items collected from various locations. For example, we will not use the sum of n data items in the algorithm for the guidance problem where n is the number of objects in the environment.

We consider several different sensors: compass to find absolute orientation, ray generator to point to a specific object, sonar or laser to measure distance to an object, rotating camera or laser to measure angles defined by pairs of point objects and the robot. We are interested in the minimal set of sensors which allow us to solve the guidance problem. Again, we only allow the robot to carry a constant number (independent of the input size of the environment) of various sensors.

This *guidance problem* can be divided into the following parts:

- the model of the map for robot navigation with a set of sensors for the map construction strategy.
- the strategy, for robot navigation from one location to another, which directs its manuevering between the obstacles.
- the observations its sensors should make to keep it on the planned track.

We want to solve the guidance problem while minimizing mechanical movement cost and measurement cost. In many cases, we make much more measurements than movements.

In Section 2, we discuss the world model and our approach in more detail. To illustrate our approach, we consider the physical enviroment to be a two dimensional plane with point objects. We show how to solve the guidance problem with several sets of sensors in Section 3. We also discuss how some of the solutions can be extended to higher dimensions. In Section 4, we discuss how to solve the guidance problem assuming even weaker sensing equipment, which can only determine the relative order of objects instead of measuring exact angles. We conclude our work in Section 5 with a discussion of future directions.

The World Model

Mobile robots may operate with imperfect sensors and imperfect mechanical control capabilities, as well as under imperfect road conditions. A result of these imperfections is that, for robots operating on a large scale space, small errors can accumulate to the extent that robots get lost. This cannot be completely resolved by improvements on these factors. Several different approaches are suggested to deal with situations where errors may occur, including approximate solutions, qualitative maps, and multi–level representation of the environment. (see, for example, (Brooks 1987; Kuipers & Byun 1991; Levitt & Lawton 1990; Teller 1992)).

A principle for resolving this problem, as suggested in the work of Levitt and Lawton (Levitt & Lawton 1990), Kuipers and Levitt (Kuipers & Levitt 1988), is to use easily recognizable features of the environment to define a global qualitative map. This global map can be built with features that can be precisely measured locally. We may view the graph world model of Dudek et al. (Dudek *et al.* 1991) as an example of this principle. In this model, nodes represent places in the environment important to the robot, and edges represent routes connecting these places. Furthermore, locally at each node, its incident edges are cyclically ordered. The robot can observe the cyclic order of the edges incident to a node only when it arrives at this node.

For the guidance problem, the first question is how to build this map; the second question is how a robot can use the map to navigate in its environment. Dudek et al. describe a method for building the map using one portable and recognizable token, which can be dropped and picked up at nodes of the graph. Once the map is built, and the position of the robot in the environment

is labelled in the map together with its orientation (i.e., an edge at this node is matched to a road at this place the robot is located), the token is no longer needed and the robot can navigate between any two nodes of the environment with the help of the map. A drawback of the graph-based model in a real environment is that it is not always straightforward to define what a node is in a robust way.

Our general approach is to consider a more realistic geometric setting where objects are located at arbitrary positions in space. We assume that the error for one sensory datum is a constant independent of the input size or complexity of the environment and so is our tolerance for accumulative errors. Therefore, to deal with the problem of accumulative errors, we allow mathematical operations only on a constant number of data items observed by the robot on different locations. We consider a point robot, though the general principle discussed in this paper would remain valid no matter what is the geometric shape of the robot. In practical problems, one would equip the robot with whatever sensors would solve the guidance problem as long as it is economically feasible. Therefore, we don't fix the sensor equipment for the robot in our discussion. Instead, we present the problem and ask what is the best set of sensors which can help us to solve it. We are interested in a robot which can carry only a constant number of sensors and would like to minimize the number of sensors.

We assume two distinguished modes for robots: stationary and moving, and introduce the following guidelines for the measurement a robot can make during each of these modes.

1. In the *stationary* mode, the robot can make precise measurements of the environment using its sensor.
2. In the *moving* mode, only qualitative changes can be observed.

This requirement makes explicit the fact that measurements can typically be made much more precise when the robot is stationary than in motion. Moreover, it takes time to analyze the data collected by sensors, so the robot may not be able to process the information quickly enough to respond to changes in the environment in real time. For example, in the stationary mode, with a rotating camera, we may measure angles between two landmarks; and we can measure distance of the robot from one landmark using laser or sonar. In the moving mode, however, the robot can do much less. The rotating camera can only keep track of the relative order of two landmarks. Laser or sonar can only tell the robot whether it is moving away from or toward a given landmark. A rotating camera can track the landmark it pointed at when it was last in stationary mode.

Three different types of cost may affect the performance of the robot: *mechanical cost*, *measurement cost*, and *computational cost*. They are usually of different orders of magnitude. Thus, we would like to obtain algorithms which minimize the vector *(mechanical cost, measurement cost, computational cost)* lexicographically. There are different metrics for these costs. For simplicity, we consider counting the number of stops the robot makes as the mechanical cost, the number of angle measurements or distance measurements as the measurement cost, and the number of basic arithemetic/logic operations as the computational cost.

Two dimensional Space with Point Objects

We consider an environment of n indistinguishable point objects in 2–dimensional space and the robot is located at an arbitrary point initially. Suppose the robot has a point object tracking sensor, that gives the robot the following abilities. In stationary mode, the robot can make precise measurements of angles between rays starting at the robot and pointing towards two arbitrary point objects. In moving mode, the robot can keep the sensor fixated on an point object, and it can count the number of point objects crossing (from left/right) the line segment between the robot and the object, or ensure no point objects cross this line segment.

First, we notice that the robot can always move to an object on the convex hull of these point objects. The idea is as follows 1. From the initial object, there are $n-1$ rays to the remaining $n-1$ objects. The robot can make a circular scan with its object tracking sensor and measure the angles formed by each pair of consecutive rays. Since the angle measurement is precise, and all angles are obtained at the same robot location, we can find the maximum angle of these $n-1$ angles. (This is a function of $n-1$ data items but collected at the same location. This would not be allowed if the data items were collected on $n-1$ different locations.) Now choose one object which delimits this maximum angle, point the object tracking sensor to it and move to this object. Continue the above operation until the maximum angle from the current robot position is greater than $> 180^o$. This implies that an object on the convex hull has been reached.

Second, once the robot reaches one object on the convex hull, it can move in a clockwise order to all the objects on the convex hull by moving each time to the point delimiting the $> 180^o$ angle that was not the

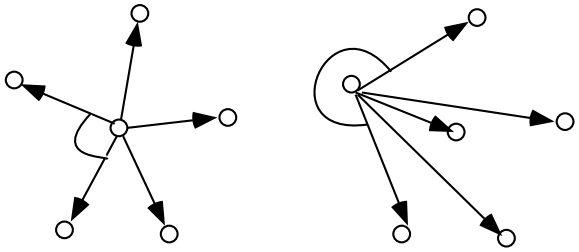

Figure 1: A robot can move to the convex hull of a set of points by repeating the following: first measure the angles between successive rays to points, and then move to a point that belongs to the largest angle. Point objects are denoted by circles. Rays are shown from the current robot position. The robot itself is not shown. The largest angle between successive rays is shown. When it becomes greater than 180 degrees, the robot is on the convex hull.

previous robot position. Theoretically, the robot can know when it has come back to the starting object on the convex hull by adding up the external angles until the sum is 360^o. However, in our model this is not allowed in order to avoid accumulative errors (these data items are collected in different locations.)

Finally, we should notice that a single object tracking sensor is not enough for the robot to construct a complete map. As a simple example, suppose we have as three objects at the vertices of an equilateral triangle and another object at its center. If we start at the center, there is no way to unambiguously match objects we see in the environment with those in the map. Even if we start at a boundary point and name objects from the object we start with, we may get lost if we ever move to the center.

Lemma 0.1 *The robot can move to a point object on the convex hull with a single point object tracking sensor in $O(n)$ moves and $O(n^2)$ angle measurements. However, in the presence of symmetries, there are situations the robot may get lost when interior objects exist.*

The guidance problem would be easy if we allowed an unbounded number of point object tracking sensors on the robot. If the robot has $n-1$ sensors, it can point each sensor to a different object while moving from one point to another. Two consecutive rays may switch their positions during the movement and the order of it depends on the movement path. Of course, the circular permutation of these rays gives the view of the robot at destination. However, for the robot to carry a number of sensors proportional to the number of objects in the environment is not realistic. Therefore, we look for different approaches to help the robot solve

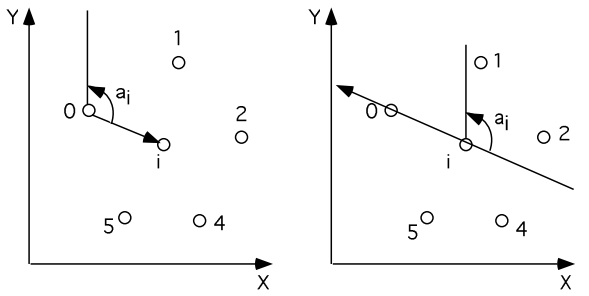

Figure 2: In the left part, angle a_i is associated with object i. In the right part, angle a_i is used to identify object i by counting the objects above the line through object i at angle a_i with respect to north. Object 0 is the reference object, i.e. the one with the smallestx coordinate.

the guidance problem with a fixed number of sensors.

In summary, we will use point objects on the convex hull as landmarks for robot navigation. Thus, we would construct a map which gives a name to each point object. At each point object on the convex hull, we have a circular permutation of all the point objects in the map which corresponds to the clockwise order they come into view. The circular permutation will be broken into a permutation in different ways according to the different sensors or landmarks we use.

The Power of an Extra Compass

Consider a coordinate system where the y axis points North. With the compass, we can move to the point with the smallest x-coordinate by testing that the north direction is tangent to the convex hull at this point in the clockwise direction. We may name this object as 0. While stationary at object 0, we may name other objects in the order they are met during a clockwise scan from the north direction 2. Each object i is associated with an angle α_i formed by the ray from 0 to i and the ray towards north (assuming no accidental alignments of objects). The number of objects in the unbounded sector between the ray towards north and the ray from 0 to i is $i-1$. Therefore, at point object i in the environment, we can figure out its name, as defined above, by drawing the straight line which forms an angle α_i with the North, and count that the number of objects bounded toward the North by this straight line is $i-1$. These operations are allowed since we only compare data collected from two different locations.

The above observation allows the robot to know the name of each object by making the above measurements once reaching the object. Starting at each object i, with the use of the compass and the above test, the

robot can visit other objects and obtain their names in a clockwise order starting from the North direction. This results in a permutation of all other objects for the rays going out of object i. For example, the permutation associated with object 3 in 2 is 12450. Thus, our map is a complete directed graph such that each node is associated with a permutation of all other nodes which correspond to the rays from this object toward other objects in the clockwise order (starting from the North). To navigate from one object to another, the robot can check the permutation of the initial node to find the position of the other node, and make a clockwise scan from the North (using the compass) until the position of the other node is reached and move toward it.

The Use of Lighthouses

We comment that a similar map can be constructed by using two lighthouses with different colours of lights. Similar to the use of compass, the position of an object is uniquely determined by the angle between the two rays from this object to the lighthouses (excluding the accidental case where an object lies on the circle defined by the two lighthouses and another object). The robot can position itself on any object on the convex hull and name other objects by its clockwise scan of them, starting from one of the lighthouses. Then, it can measure the angle formed by the two rays toward the lighthouses from any object, moving to them one by one. Notice that the robot can always move back to this initial object by first moving to one of the lighthouses. After that, the algorithm will be the same as the above except that the robot uses the angle with the lighthouses instead of the angle with the North when using a compass.

The use of object-tracking sensors

The above solutions use a set of globally available references — in the former, the compass which always points to North, in the latter, the lighthouses which are visible from anywhere. Can we construct the map and solve the guidance problem without such external help? One such solution is to use point object tracking sensors which allow the robot to walk along a straight line segment from one object to another object and to know where it comes from when reaching the destination. This can be achieved as follows, use one sensor to point to the destination and another to the object it comes from (the source). One may argue that the robot would not be able to always walk straight in our model. However, since the robot can keep track of the source object and the destination object while moving, it acts as if it has been moving on a straight line once it reaches the destination. In practice, this would be implemented by a local control strategy that keeps the robot along the line segment between the source and the destination with good approximation.

Again, the robot first moves to an object on the convex hull and names it as object 0. We name other objects based on a clockwise enumeration of the rays from object 0 toward them, starting at one ray that forms the $> 180^o$ angle and ending at the other. If the robot uses three sensors, it can always keep one pointed to object 0 and the other two to maintain the source object and the destination object. So object i is uniquely defined as follows: the number of objects bounded away from the line passing through the ray $\overline{0i}$ in its counterclockwise direction is $i-1$. The solution for constructing the map will be similar to that of using compass. The only difference will be in the rule for deciding the name of an object once the robot reaches it. To navigate from one object to another, the robot will again maintain a sensor pointed to object 0 to maintain knowledge of its orientation at all times.

Even though there is no extenal fixed reference here, the extra ray pointed to object 0 effectively creates a global reference object for the robot.

Extension to Higher Dimensions

The use of lighthouses can be naturally extended to higher dimensions by introducing one extra lighthouse for each additional dimension. Similarly, when we have several sensors, we maintain a reference to a set of fixed objects. The use of compass is usually restricted to the earthly two dimensional space. However, we may be able to obtain a fixed orientation at each point in space using different physical principles.

Robot with Qualitative Measure of Angles

In this section, we make an even weaker assumption that measurements of angles are also qualitative even when the robot is in the stationary mode. That is, the robot cannot measure exact angles but only the relative order of the objects as they are seen in a circular scan anchored at the robot's position. However, the robot can still decide whether an angle is greater than 180^o or not. Exact angle measurement would incur more cost than simply counting the number of objects in a circular scan or deciding that an angle is $> 180^o$.

With this capability, we can still decide whether the object the robot is currently positioned at a convex hull point or not.

We consider a solution using also three point object tracking sensors R_s, R_d and R_r (for tracking the source, destination and a reference point respectively). Similar results can be obtained if we use lighthouses

or a compass to substitute for some of these object tracking sensors. As noted above, with the two object tracking sensors R_s and R_d, the robot can walk along a topologically straight line segment between two point objects and hence remember the source point when reaching the destination. The following sketches the major steps to map the environment.

Step 1: Move to a convex hull point and name it object 0. This first step can no longer be done as before since we cannot decide which angle is the largest unless it is $> 180^o$. Instead, we find a line passing through the current object which minimizes the total number of objects on one side. Move to one object in the side with the fewer objects and continue until reaching an object on the convex hull which can be tested by finding an angle $> 180^o$. From now one we always let one of the three sensors lock on object 0.

Step 2: Order other points clockwise: v_{01}, v_{02}, $\cdots$, $v_{0,n-1}$, where the first and the last are the two convex hull neighbors of object 0 (they make angle $> 180^o$ around object 0). For the sake of simplifying notation, we will let i denote v_{0i}.

Step 3: For each i, $1 \leq i \leq n-1$, compute the map representation of object i as follows 3. Standing at object i, let v_{ij}, $1 \leq j \leq n-1$, denote the j-th object clockwise around i, with $v_{i1} = 0$. To establish the map, we have to find the value k such that $v_{ij} = v_{0k}$. Starting from object i we move towards j while keeping the sensor R_s locked on i, R_d locked on j, and R_r locked on 0. From j we start moving towards 0, while keeping R_s locked on j, R_d locked on 0, and R_r locked on i. When at 0, using the fact that R_s is still pointing to j, we can scan and count counterclockwise from the ray $\overline{0j}$ until v_{01}. This count is k (i.e., $v_{ij} = v_{0k}$). Using the sensors, we can come back to object i.

From the above description we formulate the following theorem.

Theorem 0.2 *With no precise angle measurements, robots with the ability (perhaps with the use of some sensors) to emulate moving on straight lines, measure* 180^o *and circularly order rays from their current position, can establish the map of n point objects with* $O(n^2)$ *moves and* $O(n^2)$ *scans as well as* $O(n^2)$ *computations.*

Discussion and Remarks

We have shown an approach for the guidance problem using imprecise data, following the world models

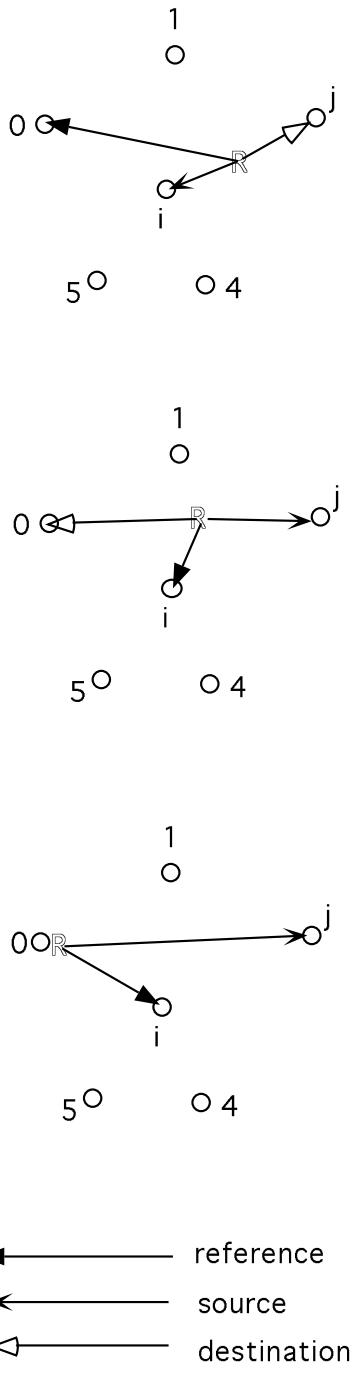

Figure 3: In the top part, the robot R moves from i to j while fixating on i,j, and 0. In the middle part, the robot moves from j to 0, while fixating on i,j, and 0. In the bottom part, the robot identifies the map representation for j by enumerating the objects swept clockwise by a ray starting just before ray $\overline{01}$ and stopping just before ray $\overline{0j}$. Object 0 is the reference object.

which provide maps with qualitative/topological information for the global environment, and quantitative/geometric information for the local environment. To illustrate our approach, we have considered the physical enviroment to be a two dimensional plane with point objects. We show how to solve the guidance problem with several sets of sensors. We are currentlly working on extending our approach when the objects are arbitrary and pairwise disjoint convex sets in the plane.

References

Blum, M., and Hewitt, C. 1967. Automata on a two dimensional tape. In *Proc. 8th IEEE Conf. on Switching and Automata Theory*, 155–160.

Blum, M., and Kozen, D. 1978. On the power of compass. In *FOCS*, 132–142.

Brooks, R. 1987. *Visual Map Making for a Mobile Robot*, volume 13. Readings in Computer Vision, edited by M. A. Fischler and O. Firschein, Morgan Kaufmann Pub., Inc., Los Altos, California.

Budach, L. 1978. Automata and labyrinths. *Math. Nachrichten* 86:195–282.

Canny, J., and Reif, J. 1987. New lower bound techniques for robot motion planning problems. In *FOCS*, 49–60.

Cox, I., and Wilfong, G. 1990. *Autonomous Robot Vehicles*. Springer Verlag, New York.

Deng, X., and Mirzaian, A. 1996. Competitive robot mapping with homogeneous markers. *IEEE Trans. on Robotics and Automation* 12:532–542.

Deng, X., and Papadimitriou, C. 1990. Exploring an unknown graph. In *FOCS*, 355–361.

Deng, X.; Kameda, T.; and Papadimitriou, C. 1991. How to learn an unknown environment. In *FOCS*, 298–303.

Dudek, G.; Jenkin, M.; Milios, E.; and Wilkes, D. 1991. Robotic exploration as graph construction. *IEEE Trans. on Robotics and Automation* 7:859–865.

Guibas, L.; Motwani, R.; and Raghavan, P. 1992. The robot localization problem in two dimensions. In *SODA 92*, 259–268.

Kuipers, B., and Byun, Y. 1991. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems* 8:47–63.

Kuipers, B., and Levitt, T. S. 1988. Navigation and mapping in large scale space. *AI Magazine* 9(2):25–43.

Levitt, T., and Lawton, D. T. 1990. Qualitative navigation for mobile robots. *Artificial Intelligence* 44:305–360.

Rivest, R., and Schapire, R. 1989. Inference of finite automata using homing sequences. In *STOC*, 411–420.

Teller, S. 1992. Visibility computations in densely occluded polyhedral environments.